INFORMATION MANAGEMENT FOR A LARGE MULTIDISCIPLINARY PROJECT

Kennie H. Jones
NASA Langley Research Center
Hampton, VA 23681

Donald P. Randall
NASA Langley Research Center
Hampton, VA 23681

Catherine K. Cronin
NASA Langley Research Center
Hampton, VA 23681

Abstract

In 1989, NASA's Langley Research Center (LaRC) initiated the High-Speed Airframe Integration Research (HiSAIR) Program to develop and demonstrate an integrated environment for high-speed aircraft design using advanced multidisciplinary analysis and optimization procedures. The major goals of this program were to evolve the interactions among disciplines and promote sharing of information, to provide a timely exchange of information among aeronautical disciplines, and to increase the awareness of the effects each discipline has upon other disciplines. LaRC historically has emphasized the advancement of analysis techniques. HiSAIR was founded to synthesize these advanced methods into a multidisciplinary design process emphasizing information feedback among disciplines and optimization. Crucial to the development of such an environment are the definition of the required data exchanges and the methodology for both recording the information and providing the exchanges in a timely manner. These requirements demand extensive use of data management techniques, graphic visualization, and interactive computing. HiSAIR represents the first attempt at LaRC to promote interdisciplinary information exchange on a large scale using advanced data management methodologies combined with state-of-the-art, scientific visualization techniques on graphics workstations in a distributed computing environment. The subject of this paper is the development of the data management system for HiSAIR.

Background

NASA's Langley Research Center (LaRC) has, since its beginning, maintained a leadership role in research for the advancement of techniques to analyze aircraft designs. In recent years, aircraft research has evolved into distinct disciplines: aerodynamics, structures, propulsion, performance, controls, and others. With the distinction came isolation, as each discipline concentrated on activities related to its own concerns. This isolation has created at least two problems in the incorporation of advancing analysis techniques into the design process. Disciplines have come to view production of their analysis results as a final product. For example, a Computational Fluid Dynamics analyst may consider his task complete when a given geometric model has been analyzed under a given set of flight conditions to calculate a distribution of pressures. He may therefore view his job as the improvement of methods used to calculate the pressures without consideration of the effect on the design process as a whole. Such an isolation of purpose could result in an optimal aerodynamic design that can not contain a sufficient supportive structure. Secondly, a discipline may require data from another discipline as input to an analysis, but may develop or acquire capabilities to generate these data locally. While this approach maintains independence from a remote group, he may not benefit from their expertise or their most recent improvements.

In 1989, LaRC's Director, Dr. Richard Peterson, initiated the Multidisciplinary Research Advisory Committee (MRAC). This committee was born from the observation of this isolation of disciplines and a desire to ensure incorporation of the most advanced methods in the design of aircraft. Soon after its inception, the MRAC formed the High-Speed Airframe Integration Research (HiSAIR) program. HiSAIR's charter is to develop and demonstrate an integrated environment for high-speed aircraft design using advanced multidisciplinary analysis and optimization procedures. This is to be accomplished through three stages: establishment of multidisciplinary analysis processes providing unprecedented levels of discipline fidelity and turn-around time, development of a local optimization design process using this analysis capability, and development of global optimization methods providing full simulation of the design process. A major goal of this program was to develop the interactions among disciplines and promote sharing of information. Information sharing reduces discipline isolation and encourages the use of the most advanced techniques while increasing the awareness of the effects each discipline has upon other disciplines in the design process. The HiSAIR community was formed from selected representatives of each of LaRC's research directorates and required support organizations.

Crucial to the development of such an environment are the definition of the required data exchanges and the methodology for both recording the information and providing the exchanges in a timely manner. New information generated by any discipline must be quickly available to all other disciplines and any proposed modification to the design must be quickly evaluated for its effect on other disciplines and the design as a whole. Early in the design process, the amount of data is relatively small and the required exchanges simple enough that informal transfers are feasible. However, as the design progresses, data is produced in such volume and the required exchanges increase in complexity to a point that more formal methods of data storage, retrieval, and visualization become a necessity. New information generated within a discipline must be transmitted quickly and accurately to all those affected. Transmissions must be accomplished with minimal impact to the researcher and are most effective if the methods used actually decrease the workload. These requirements demand extensive use of data management techniques, graphic visualization, and interactive computing. HiSAIR represents the first attempt at LaRC to promote interdisciplinary information exchange on a large scale using advanced data management methodologies combined with state-of-the-art scientific visualization techniques on graphics workstations in a distributed computing environment.

The subject of this paper is the development of the data management system for HiSAIR. The first step was to gain an understanding of the analysis processes within each discipline and document the required input and resulting output data. An artificial intelligence system developed at LaRC was used to analyze the data flow between disciplines and graph the resultant dependencies.[4] A data tracking system was then designed to provide a common method for recording data in such a way that other researchers can quickly and easily locate and review data of interest to their analysis process. When data is found that can serve as input to a researcher's analysis, it can be transferred from its storage location on the distributed network to the researcher's local computer for analysis. Resulting output can then be submitted to the data tracking system making it available to other researchers.

A critical factor in the development of a data management system for aircraft design is the methodology for development, transference, and modification of geometry. Because of the large number of analytical iterations required, the geometry must be created and modified quickly. In a multidisciplinary design environment, many different representations of the geometry must be created to support different analyses. Prior to HiSAIR, each discipline created a geometric model of the aircraft using its own tools and techniques. A major goal of the data management effort within HiSAIR is to consolidate the geometry generation into a single software tool that can output the various required geometric representations from a common model.

Because disciplines have evolved their analysis techniques on different computing systems, their procedural methods for executing analysis programs differ substantially. Often the methods are highly operating system dependent and use such methods as editing potentially sharable data files using text editors. Another major goal of the data management effort within HiSAIR has been to provide a Graphical User Interface (GUI) for program execution and data review/modification that presents a consistent computer interface across disciplines and is independent of the operating system.

Approach

The first step in the design of a data management system to support a multidisciplinary design environment is to survey the contributing disciplines to assess the requirements for data exchange. The initial plan for HiSAIR was to construct a survey form. The survey form was designed to provide an understanding of the analysis programs in use, their required input data, and their resulting output data. The form was distributed to representatives of each discipline with instructions to collect responses from their team members and return the forms. Upon receiving the forms, it was obvious to the Data Management Team that information was missing. Two subsequent revisions of the form were redistributed and analyzed followed by personal interviews with each discipline team by members of the Data Management Team.

The resulting information was organized and used as input to the DEMAID program.[4] Based on the input and output of each program, DEMAID determines the order in which the programs must be executed. Figure 1 illustrates the output from the DEMAID analysis. The rectangles arranged along the diagonal represent the order of program execution determined by DEMAID. Horizontal lines connected to the right side of a rectangle indicate output from the program represented by that rectangle. Vertical lines connected to the top of a rectangle indicate input to the program represented by the rectangle. A filled circle at an intersection indicates that output from a program (to the left of the horizontal line) is required as input to another program (at the bottom of the vertical line). Horizontal and vertical lines below the diagonal represent feedback loops where data is sent back to a previously executed program requiring re-execution of at least that program. Notice that the program to create geometry (upper left corner) must be executed before any further analysis can commence.

The chart in Figure 2 depicts another view of the data flow among the HiSAIR disciplines. Here the filled circles have been replaced with named data sets. Unlike the DEMAID output, the required execution paths are not obvious. Notice that the rectangles represent analysis capability that can be provided by programs of differing levels of fidelity. For example, aerodynamic pressures may be computed using linear algorithms or higher order analyses such as Euler or Navier-Stokes methods. Structural analysis can be accomplished using finite element methods or simpler flat plate methods. A major goal of the HiSAIR project is to incorporate higher order analysis methods wherever possible and to assess when simpler methods are sufficient.

After examining the data flow required to facilitate a baseline multidisciplinary capability, the methods used to store, manipulate, and organize the data were analyzed. With one exception, all researchers relied upon flat files to contain both input and output data for their programs. Although some data was stored in binary files, most files were ASCII. Input files were usually created or modified using a text editor. As interactions between disciplines began to develop, much "hand work" was required to transfer data from one discipline to another. Output files from one discipline were transferred to another discipline where subsets of the data were extracted and modified using a text editor. In some cases, files were transferred on paper. The receiving researcher read the paper and typed the useful data into his input file. The one exception to these informal methods was found among researchers developing structural optimization techniques. They had constructed a system of programs including aerodynamic analysis, flat plate structural analysis, and structural optimization that communicated through a common data base.[1] Although the data flowed well between these programs, the tools for reviewing and modifying data were limited.

The early attempts to communicate data were useful to demonstrate the potential benefits of multidisciplinary analysis, but it was obvious that for these methods to become practical in a design system, several improvements were necessary. First, better methods were required to manage the files. Most researchers depended on file names to remind them of the source of the data. Usually a file name is a short string of characters and is insufficient to convey enough information to document the significance of the data. Consequently, the researcher uses the file name to link the file to other information often retained only in his memory. As an example, a file may contain a set of geometric points and the aerodynamic pressures calculated at those locations. However, other information is important to the proper interpretation of the data such as the altitude and speed at which the aircraft was analyzed. This related information is often described as "meta data." Not only is it unreliable to depend upon human memory to store meta data but it also requires that the person responsible for the data remain accessible for its proper retrieval. In one case, a researcher added header records to his file with the text editor to store meta data. Although this method was superior to memorizing the meta data and the file name with which it was associated, it required the researcher to enter the editor to search for a particular file. This methodology may be sufficient when there are few files, but in a design environment where many iterations of analysis may produce an abundance of files, it would be prohibitively time-consuming to edit each file. Clearly, a better method for organizing and locating data could protect the integrity of the data by reducing the dependence on human memory while also improving the timely exchange of information. It was recognized that successful implementation of an improved methodology must result in a system for reliable information management that decreased the work load of the researcher, not one that presented an additional burden.

Secondly, better methods were required to review and modify data and execute programs. Most of the HiSAIR engineers were comfortable using text editors to view and modify data and some liked the flexibility offered by their favorite editor. However, a text editor offers potentially harmful opportunity, ranging from the ability to corrupt the data values to the ability to corrupt the format of the file rendering it unreadable by the analysis program. A more structured approach to data review and modification would provide protection from potential corruption, and better tools could improve the speed at which modifications are made. Throughout the HiSAIR community, researchers executed programs directly from commands of the host computer's operating system. This placed an undue burden on the researcher to become familiar with the operating system. In one case, a discipline converted from a VAX/VMS computer system to UNIX workstations which required the researchers to take time from their research to learn a new operating system.

Thirdly, a more coordinated methodology was required for geometry generation. A conceptual model was initiated as a coarse wire frame model, a finite set of points on the surface of the aircraft. This model was passed to a geometry discipline to generate a fine mesh grid for higher-order aerodynamic analysis. The structural analyst read the coarse model into an application to generate an approximate set of surface patches representing the aircraft. Other disciplines used a two dimensional drawing of the aircraft planform to divide it into sections for flat plate structural analysis or aeroservoelastic analysis. These three methods created a divergence of the geometry and could result in the analysis of different aircraft.

To provide a solution to these three identified requirements, a HiSAIR Data Management (HDM) system was designed. Depicted in Figure 3, HDM provides a Window System (X) and OSF/Motif Graphical User Interface (GUI) interfaced with a commercially available Data Base Management System (DBMS). It consists of three major subsystems: the data tracking system, a geometry generator, and an application execution environment. The data tracking system is described in detail below. Because the components of the geometry generator [] and application execution environment [] are well documented, discussion of these subsystems will be restricted to their use within the HiSAIR program.

HDM Data Tracking System

The heart of HDM is its data tracking system, designed to provide a well-defined and easy-to-use method for sharing technical data among research disciplines. The system allows researchers to provide results to others within and across disciplines and to locate data sets that will be useful to their research. It facilitates communication with the originator of the data set, as well as transfer of that set to others. It consists of data base, GUI software, and a set of rules governing that software. The software provides an automated method of storing and retrieving information about specific data sets and the ability to locate them. The rules define the privileges each person may have to retrieve and store data.

The data tracking system provides an automated means of storing, identifying, modifying, and retrieving data sets from a data base and a designated file storage area. The data base may contain the actual data along with various meta data used to describe the actual data. Alternatively, meta data may only be stored within the data base while the actual data is stored in a file outside the data base. Part of the meta data in this case includes the location of the file. A prime directive in the design of the data tracking system was that it should not be limited to the data sets of one research area, but shall include sets from multiple disciplines and provide extensibility to include unforeseen data sets. The data tracking system provides a point-and-click user interface, utilizing an interactive forms capability. The program operates such that the user is unaware of the specific DBMS used to store data or the language used to retrieve data. The user defines a query of the data base by completing a form on the display. The data base query statement is then generated internally within the program. Upon completion of a search, the program displays the results to the terminal screen in a format described by the user through the interactive form. A key feature of the program is the ability to save the query/output forms for later reuse. Once a query/output form has been established, the user can open the form, quickly modify query values, and search for the data. The value in this approach is in the assumption that similar queries will be often executed by users. After a search, the user may peruse the displayed data or request that a copy of the results be sent to a printer. In a future release, the user may request a chosen data set to be downloaded to a specific computer.

Functions provided by the HDM data tracking system can be classified into three categories: System Administration, Data Base Administration, and Data Base Query. Users are assigned access to these functions based on their needs and interface with these functions through the use of interactive forms and menus. System Administration provides the capability to control user groups and their access to other functions. A list of the users that have access to the HDM software and their associated passwords is maintained in the data base. Users are assigned to one or more groups. Groups are meant to be formed of users that share a common interest in data access. Privileges (i.e., add to the data base, query the data base, delete from the data base, add users, modify groups, etc.) are maintained for each user and group. Data Base Administration provides the capability of adding entries to the data base. Data Base Query provides capabilities of identifying one or more data sets that meet the criteria specified by the user. In addition to the choice of a set of default query/output formats, the user has the capability of creating an ad hoc query by using a “palette” of search criteria. The user may choose any of the search criteria available in the palette to custom design his own form. Likewise, the user will be able to define the format of the records returned from the query by using the same method. By using the output format's palette, the user is also able to customize which data are returned from the data base query.

Note that the system is designed to provide functionality, not dictate policy. That is, formation of groups is controlled by the HiSAIR community and privileges may be assigned along a continuum ranging from all users have all privileges to a strictly controlled access. A suggested approach for HiSAIR is to create a group for each discipline and to assign a small number of administrators with System Administration privileges, assign a small number of members for each group with Data Administration privileges, and permit Data Base Query privileges by all users.

The meta data for a given data set is clearly divided into two subsets: data independent and data dependent. Data independent information consists of meta data that should be stored regardless of the source or format of the tracked data. Examples of this include the configuration, creation date, point-of-contact name, program that created the data, file name and location, description, etc. Data dependent information consists of meta data that is specific to a particular data set. For example, if the data set to be tracked is aerodynamic pressures, the altitude and speed used in the analysis should be stored as well as the pressures. In the HDM data tracking system, data dependent information is identified through the File Type and File Class attributes. It is important to note that the meta data associated with a specific data set is defined in terms of a data model which is determined jointly by the Data Management Team and the researchers responsible for this data set. The attributes File Type and File Class are important in defining this data set specific data model. In addition, these attributes are used to restrict search criteria and query output in order to permit only realistic and legitimate queries.

Querying Using the HDM Data Tracking System

After entering the program by supplying a validated user name and password, the Control Window in Figure 4 appears on the display screen. The pulldown menu resulting from the “File” selection provides only an “Exit” selection to terminate the program. The “Change” pulldown menu provides functions for System Administration. All users can change their password and review user or group information using these functions. Only users with System Administration privileges may change user or group information. Selection of the “Actions” pulldown menu provides access to the displayed functions. “Add to Data Base,” “Modify the Data Base,” and “Delete from Data Base” selections represent Data Base Administration functions which require Data Base Administration privileges. Selection of “Query the Data Base” results in the display of the HDM Query Control Window in Figure 5. The pulldown menu under “File” provides selections to return to the main menu and to print the display screen. Selection of “QuerySetup” results in the pulldown menu displayed. These functions allow the creation (“New”) of a new query form or reclamation of a previously defined query form (“Open”). The window entitled “Query Form” in Figure 6 depicts the result of opening a previously defined form. Notice that the entities within the window form the conditional clause for a query, “where Last Name equals Coen or equals Fenbert and File Class equals Output and File Type equals FLOPS.” The conditional clause may be used as is or the attribute values (“Coen,” “Fenbert,” “Output,” or “FLOPS”) can be changed before execution of the search. A check box to the right of the value indicates that a list of allowable selections is available. Selecting the check box results in a display of the selectable list of values for this attribute. This methodology reduces the number of required key strokes and the chance of a typing error. It also reduces the dependence on user memory to recall acceptable values. Proven to provide a great productivity improvement, this methodology is used throughout the program where applicable. Other parts of the conditional clause can be changed as well. Selection of the command buttons containing logical operators (“EQ”) will result in the display of an option list which allows modification of the boolean operator. Selection of the “Skull and Crossbones” icon beside a boolean expression will remove it from the clause. Selection of the pulldown menu “Show Palette” below the “Edit” menu results in the display of the “Query Palette” window (see Figure 6) which contains a selectable list of general and data set specific meta data attribute names. Selection of an attribute name results in the addition of a boolean expression to the conditional clause. Notice that all but the last two attributes are “data independent.” If File Type and File Class are selected for inclusion in the conditional clause, their values will determine the picklist entries of “data-dependent” attributes (in this figure, “Polar Plot” and “Polar Plot Altitude” are attributes dependent on the File Class “Output” and the File Type “FLOPS”). Also notice that the Query Palette, in this case, is divided into two separate sections with independent scroll bars controlling the two separate picklists.

Selection of “Define Output...” on the “Query Form” toggles the display to the Results Form entitled “Query Form” depicted in Figure 7. Using similar techniques to those described above, the attributes to be displayed and the order in which they appear in the Results window can be controlled. Selection of the pulldown menu “Show Palette” below the “Edit” menu results in the display of the “Query Palette” allowing the selection of attributes to be added to the display list. The “Query Form” window lists the results of the palette selections. Selections of the command buttons to the right of the attribute name results in the display of an option list of sorting methods (NONE, ASCENDING, or DESCENDING) for sorting the rows returned in the search.

When the conditional clause and output requirements are acceptable, the “Start” pulldown menu under the “Search” menu on the Query Control Window may (Figure 5) may be selected to execute the query. From the pulldown menu under the “Results” menu, the user can opt to display the results to the screen, save them to a file, or print them. Figure 8 demonstrates the format of the displayed output. Having a spreadsheet appearance, the attribute names head the columns. The window can be resized or scrolled to reveal large data sets. Cells that contain more data than can be reasonably displayed in the results window can be expanded (Figure 9). The use of the cell expansion capability is data model dependent. This feature may be used to accommodate lengthy meta data text attributes such as a file description, but may also be employed to examine subsets of the actual data.

Adding Data Using the HDM Data Tracking System

Adding a data set to the data base follows the same methodology as the Query functions. Users with add privileges can select the “Add to Data Base” menu (Figure 4) resulting in the Add Control Window depicted in Figure 10. From the pulldown menu under “InputForm” the user can open a “New” input form or a “User” form. A new form has no default values. A user form has any default values defined at the time it was saved. Using a user form can save typing or picking values that are often unchanged. Following the opening of a form, a File Type and File Class must be selected resulting in the display of the Add Form depicted in Figure 11. The Add Form contains all of the attributes that must be defined for the selected data set. Attribute values can be entered from the keyboard or, for those that have check boxes, selected from a picklist. Depending on the data model, the Add Form may have more than one section as indicated by the “Previous Section” and “Next Section” command buttons. All sections of this form must be completed before the data set is allowed to be added. When all values have been defined, the data set can be stored by selecting the “Add to Data Base” pulldown menu under the “Actions” menu on the Add Control Window (Figure 10).

Application Execution Environment

The Environment for Application Software Integration and Execution (EASIE) is a methodology and a set of software utility programs developed at the LaRC for coordinating the use of engineering design and analysis computer programs.[2] Under user direction, EASIE controls the execution of independently developed programs and manages the flow of data to and from a common relational data base in order to accomplish design or analysis objectives. Among the tools provided are an executive controller for selecting a data base and executing programs and a program facilitating the review and modification of program input and output data. These tools were presented to the HiSAIR community as a useful approach to reducing the reliance on text editors for data modification and dependence on the computer operating system for program execution. However, the tools were based on older technology and were driven by a command interface. The researchers, already appreciating the advantages to be offered through windowing interfaces and full screen available on workstations, balked at the thought of using a interface that they considered a step backwards in capability. Plans were already forming to convert the EASIE system to the X Window System and OSF/Motif environment. The executive and review programs have been converted and are available to those HiSAIR disciplines choosing this approach within their own organizations. Use of this environment will provide a consistent user interface among the discipline application systems but is not a prerequisite for using the data tracking system.

Geometry Generation

The Solid Modeling Aerospace Research Tool (SMART) was developed by the Vehicle Analysis Branch (VAB) at LaRC to aide in the construction of analysis models of space transportation vehicles.[3] Based on Bezier curves and surfaces, SMART appeared to have promise as a common tool for generating geometric models of high-speed aircraft. Key features of the program were the ability to construct a wing surface from parametric descriptions and the ability to easily construct internal structural components for wings, tanks, etc. GEOLAB, a laboratory led by a team of grid generation experts in the Analysis and Computation Division (ACD) at LaRC were progressing towards enhancements to support development of models of sufficient fidelity to be used as input to higher order aerodynamic analysis systems. A consortium consisting of representatives from several HiSAIR disciplines and the SMART developers was established to specify other enhancements that would make SMART a more useful tool for the consolidation of geometric models used by the HiSAIR community. While SMART had the initial capabilities to generate wing surfaces and internal structures for wings, enhancements were identified that will improve the usefulness of these features to HiSAIR including plans to extend the structural definition function to fuselages. With these enhancements completed, SMART will provide a common geometry generation tool for the development of the various modeling representations.

Current Status

The initial implementation of the HDM data tracking system is complete for the System Administration and Data Base Query functions. The Data Base Administration functions are currently accessible through an interim character based application specific to each implemented data set. The GUI to these functions is scheduled to be completed soon. Data sets from selected disciplines, including Geometry, Performance, Aerodynamics, Aeroservoelasticity, and Controls have been implemented. The data tracking system is in use and under review by representatives of these organizations. Results retrieval is currently limited to screen display, downloading to a local file, or printing. Future plans call for the capability to download the data directly from the shared data base server to the users local computer.

The GUI version of EASIE has recently been completed and is currently under review by several organizations. Plans are to use this system for those organizations that may benefit from its executive controller and data review capability. EASIE also provides data management tools that are to be used in the implementation of the data download capability planned for the data tracking system.

The grid generation enhancements to SMART have been completed and are in use by GEOLAB. The initial implementation of the aerodynamic enhancements for SMART have been completed and are under review by selected HiSAIR participants. The structural enhancements are under development but should be completed soon.

Conclusion

These data management methodologies have been designed to make data more accessible among disciplines, promote the sharing of data across disciplines, and provide a consistent user interface for data exchange and program execution while decreasing the time required and improving reliability for such exchanges. The data tracking system promises to provide a useful means of controlling configurations and managing the flow of data among disciplines. It also will be useful for management of data within a discipline as it provides tools to document the data. The facilities for group formation and privilege assignment provide the flexibility to control the input and modification of data sets. The design of the data tracking system was such that its use is not specific to HiSAIR which promises potential use in other applications requiring control of data flow.

Use of EASIE may prove beneficial to some, if not all, HiSAIR disciplines, but is not mandatory for use of the data tracking system. EASIE has been useful in several applications at LaRC and in industry. The new GUI has greatly improved its effectiveness.

The future of SMART is somewhat uncertain at this point. HiSAIR shares a dilemma with much of NASA and industry: use commercial software for the support and development resources commercial companies have to offer or develop local software customized to specific needs. SMART was originally conceived and implemented at a time when no commercial software evaluated met the needs of the VAB user community. Since that time, commercial systems have improved and are providing more utilities for local customization. The current plan is to complete the recent enhancements to SMART and freeze a version for use in its current state. Then a decision will be made to continue local support of SMART or search for an acceptable alternative in the commercial market.

References

1. A. R. Dovi, G. A. Wrenn, J.F.M Barthelemy, P. G. Coen, L. E. Hall, “Mult-D Design Integration System for a Supersonic Transport Aircraft,” *AIAA 92-4841*, September 1992.

2. K. H. Jones, D. P. Randall, L. F. Rowell, R. L. Gates, C. M. Nichols, S. B. Williams, “Environment for Application Software Integration and Execution,” *Proceedings of the ASME International Computers in Engineering Conference and Exposition*, August 1992.

3. M. L. McMillian, J. J. Rehder, A. W. Whilhite, J. L. Swing, J. Spangler, J. C. Mills, “A Solid Modeler for Aerospace Vehicle Design, *AIAA 87-2901*, September 1987.

4. J. Rogers, “A Knowledge-Based Tool for Multilevel Decomposition of a Complex Design Problem,” *NASATP 2903*, May 1989.

Figure Legend:

1. Output from DEMAID (not available on-line)
2. HiSAIR Data Flow (not available on-line)
3. HiSAIR Data Management (HDM) System
4. HDM Control Window
5. HDM Query Control Window
6. HDM Query Form
7. HDM Output Form
8. HDM Results Output
9. HDM Cell Expansion
10. HDM Add Control Window
11. HDM Add Form

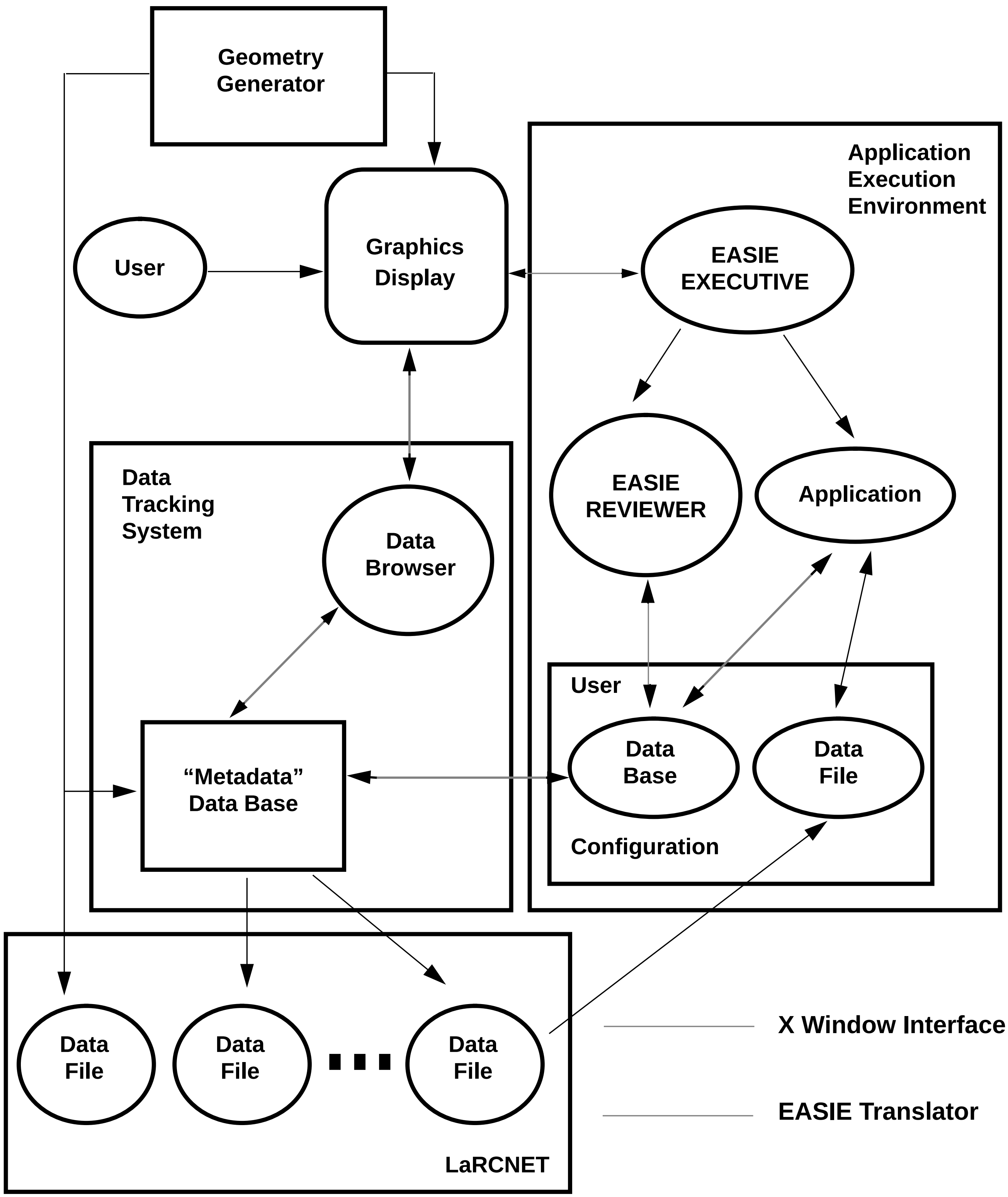

Figure 3. HiSAIR Data Management (HDM) System

Figure 4. HDM Control Window

Figure 5. HDM Query Control Window

Figure 6. HDM Query Form

Figure 7. HDM Output Form

Figure 8. HDM Results Output

Figure 9. HDM Cell Expansion

Figure 10. HDM Add Control Window

Figure 11. HDM Add Form